Optimal Capacitor Placement in a Radial Distribution System using Shuffled Frog Leaping and Particle Swarm Optimization Algorithms

Saeid Jalilzadeh[1] , M.Sabouri[2] , Erfan.Sharifi[3]
Zanjan University, Zanjan, Iran[1,2], Azad university of Miyaneh branch,Iran[3]
Jalilzadeh@znu.ac.ir[1] , M.Sabouri@znu.ac.ir[2] e.sharify@gmail.com[3]

Abstract—**This paper presents a new and efficient approach for capacitor placement in radial distribution systems that determine the optimal locations and size of capacitor with an objective of improving the voltage profile and reduction of power loss. The solution methodology has two parts: in part one the loss sensitivity factors are used to select the candidate locations for the capacitor placement and in part two a new algorithm that employs Shuffle Frog Leaping Algorithm (SFLA) and Particle Swarm Optimization are used to estimate the optimal size of capacitors at the optimal buses determined in part one. The main advantage of the proposed method is that it does not require any external control parameters. The other advantage is that it handles the objective function and the constraints separately, avoiding the trouble to determine the barrier factors. The proposed method is applied to 45-bus radial distribution systems.**

Index Terms—**Distribution systems, Capacitor placement, loss reduction, Loss sensitivity factors, SFLA, PSO**

I. INTRODUCTION

The loss minimization in distribution systems has assumed greater significance recently since the trend towards distribution automation will require the most efficient operating scenario for economic viability variations. Studies have indicated that as much as 13% of total power generated is wasted in the form of losses at the distribution level [1]. To reduce these losses, shunt capacitor banks are installed on distribution primary feeders. The advantages with the addition of shunt capacitors banks are to improve the power factor, feeder voltage profile, Power loss reduction and increases available capacity of feeders. Therefore it is important to find optimal location and sizes of capacitors in the system to achieve the above mentioned objectives. Since, the optimal capacitor placement is a complicated combinatorial optimization problem, many different optimization techniques and algorithms have been proposed in the past. Schmill [2] developed a basic theory of optimal capacitor placement. He presented his well known 2/3 rule for the placement of one capacitor assuming a uniform load and a uniform distribution feeder. Duran *et al* [3] considered the capacitor sizes as discrete variables and employed dynamic programming to solve the problem. Grainger and Lee [4] developed a nonlinear programming based method in which capacitor location and capacity were expressed as continuous variables. Grainger *et al* [5] formulated the capacitor placement and voltage regulators problem and proposed decoupled solution methodology for general distribution system. Baran and Wu [6, 7] presented a method with mixed integer programming. Sundharajan and Pahwa [8], proposed the genetic algorithm approach to determine the optimal placement of capacitors based on the mechanism of natural selection. In most of the methods mentioned above, the capacitors are often assumed as continuous variables. However, the commercially available capacitors are discrete. Selecting integer capacitor sizes closest to the optimal values found by the continuous variable approach may not guarantee an optimal solution [9]. Therefore the optimal capacitor placement should be viewed as an integer-programming problem, and discrete capacitors are considered in this paper. As a result, the possible solutions will become a very large number even for a medium-sized distribution system and makes the solution searching process become a heavy burden. In this paper, Capacitor Placement and Sizing is done by Loss Sensitivity Factors and Shuffled Frog Leaping Algorithm (SFLA) respectively. The loss sensitivity factor is able to predict which bus will have the biggest loss reduction when a capacitor is placed. Therefore, these sensitive buses can serve as candidate locations for the capacitor placement. SFLA is used for estimation of required level of shunt capacitive compensation to improve the voltage profile of the system. The proposed method is tested on 45 bus radial distribution systems and results are very promising. The advantages with the shuffled frog leaping algorithm (SFLA) is that it treats the objective function and constraints separately, which averts the trouble to determine the barrier factors and makes the increase/decrease of constraints convenient, and that it does not need any external parameters such as crossover rate, mutation rate, etc.

The remaining part of the paper is organized as follows: Section II gives the problem formulation; Section III sensitivity analysis and loss factors; Sections IV gives brief description of the shuffled frog leaping algorithm; Section V develops the test results and Section VI gives conclusions.

II. PROBLEM FORMULATION

The real power loss reduction in a distribution system is required for efficient power system operation. The loss in the system can be calculated by equation (1) [10], given the system operating condition,

$$P_L = \sum_{i01}^{n} \sum_{i01}^{n} A_{ij}(P_iP_j + Q_iQ_j) + B_{ij}(Q_iP_j - P_iQ_j) \quad (1)$$

© 2012 ACEEE
DOI: 01.IJNS.03.02.72

Where,
$$A_{ij} = \frac{R_{ij}\cos(\delta_i - \delta_j)}{V_i V_j}$$
$$B_{ij} = \frac{R_{ij}\sin(\delta_i - \delta_j)}{V_i V_j}$$

where, *Pi* and *Qi* are net real and reactive power injection in bus *'i'* respectively, *Rij* is the line resistance between bus *'i'* and *'j'*, *Vi* and *δi* are the voltage and angle at bus *'i'*.

The objective of the placement technique is to minimize the total real power loss. Mathematically, the objective function can be written as:

Minimize: $$P_L = \sum_{k=1}^{Nsc} Loss_k \qquad (2)$$

Subject to power balance constraints:

$$\sum_{i=1}^{N} Q_{Capacitori} = \sum_{i=1}^{N} Q_{Di} + Q_L \qquad (3)$$

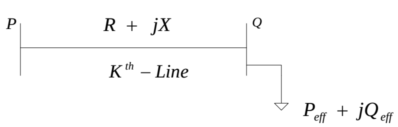
Voltage constraints: $$|V_i|^{\min} \le |V_i| \le |V_i|^{\max} \qquad (4)$$

Current limits: $$|I_{ij}| \le |I_{ij}|^{\max} \qquad (5)$$

where: $Loss_k$ is distribution loss at section k, NS_C is total number of sections, P_L is the real power loss in the system, $P_{Capacitori}$ is the reactive power generation Capacitor at bus i, P_{Di} is the power demand at bus i.

III. SENSITIVITY ANALYSIS AND LOSS SENSITIVITY FACTORS

The candidate nodes for the placement of capacitors are determined using the loss sensitivity factors. The estimation of these candidate nodes basically helps in reduction of the search space for the optimization procedure.Consider a distribution line with an impedance *R*+*jX* and a load of P_{eff} + jQ_{eff} connected between '*p*' and '*q*' buses as given below.



Active power loss in the k^{th} line is given by, $[I_k^2]*R[k]$ which can be expressed as,

$$P_{lineloss}[q] = \frac{(P_{eff}^2[q] + Q_{eff}^2[q])\,R[k]}{(V[q])^2} \qquad (6)$$

Similarly the reactive power loss in the k^{th} line is given by

$$Q_{lineloss}[q] = \frac{(P_{eff}^2[q] + Q_{eff}^2[q])\,X[k]}{(V[q])^2} \qquad (7)$$

Where, $P_{eff}[q]$ = Total effective active power supplied beyond the node '*q*'.

$Q_{eff}[q]$ = Total effective reactive power supplied beyond the node '*q*'.

Now, both the Loss Sensitivity Factors can be obtained as shown below:

$$\frac{\partial P_{lineloss}}{\partial Q_{eff}} = \frac{(2*Q_{eff}[q]*R[k])}{(V[q])^2} \qquad (8)$$

$$\frac{\partial Q_{lineloss}}{\partial Q_{eff}} = \frac{(2*Q_{eff}[q]*X[k])}{(V[q])^2} \qquad (9)$$

Candidate Node Selection using Loss Sensitivity Factors:

The Loss Sensitivity Factors ($\partial P_{lineloss} / \partial Q_{eff}$) are calculated from the base case load flows and the values are arranged in descending order for all the lines of the given system. A vector bus position 'bpos[*i*]' is used to store the respective 'end' buses of the lines arranged in descending order of the values ($\partial P_{lineloss} / \partial Q_{eff}$).The descending order of ($\partial P_{lineloss} / \partial Q_{eff}$) elements of "bpos[i]' vector will decide the sequence in which the buses are to be considered for compensation. This sequence is purely governed by the ($\partial P_{lineloss} / \partial Q_{eff}$) and hence the proposed 'Loss Sensitive Coefficient' factors become very powerful and useful in capacitor allocation or Placement. At these buses of 'bpos[*i*]' vector, normalized voltage magnitudes are calculated by considering the base case voltage magnitudes given by (norm[*i*]=*V*[*i*]/0.95). Now for the buses whose norm[i] value is less than 1.01 are considered as the candidate buses requiring the Capacitor Placement. These candidate buses are stored in 'rank bus' vector. It is worth note that the 'Loss Sensitivity factors' decide the sequence in which buses are to be considered for compensation placement and the 'norm[*i*]' decides whether the buses needs *Q*-Compensation or not. If the voltage at a bus in the sequence list is healthy (i.e., norm[*i*]>1.01) such bus needs no compensation and that bus will not be listed in the 'rank bus' vector. The 'rank bus' vector offers the information about the possible potential or candidate buses for capacitor placement. The sizing of Capacitors at buses listed in the 'rank bus' vector is done by using Shuffled Frog Leaping Algorithm.

IV. OPTIMIZATION METHPDS

A. Shuffled frog leaping algorithm

The SFLA is a meta heuristic optimization method that mimic the memetic evolution of a group of frogs when seeking for the location that has the maximum amount of available food. The algorithm contains elements of local search and global information exchange [11]. The SFLA involves a population of possible solutions defined by a set of virtual frogs that is partitioned into subsets referred to as memeplexes. Within each memeplex, the individual frog holds ideas that can be influenced by the ideas of other frogs, and the ideas can evolve through a process of memetic evolution. The SFLA performs simultaneously an independent local search in each memeplex using a particle swarm optimization like method. To ensure global exploration, after a defined number of memeplex evolution steps (i.e. local search iterations), the virtual frogs are shuffled and reorganized into new memeplexes in a technique similar to that used in the shuffled complex evolution algorithm. In addition, to provide the opportunity for random generation of improved

© 2012 ACEEE
DOI: 01.IJNS.03.02.72

information, random virtual frogs are generated and substituted in the population if the local search cannot find better solutions. The local searches and the shuffling processes continue until defined convergence criteria are satisfied. The flowchart of the SFLA is illustrated in fig. 1.

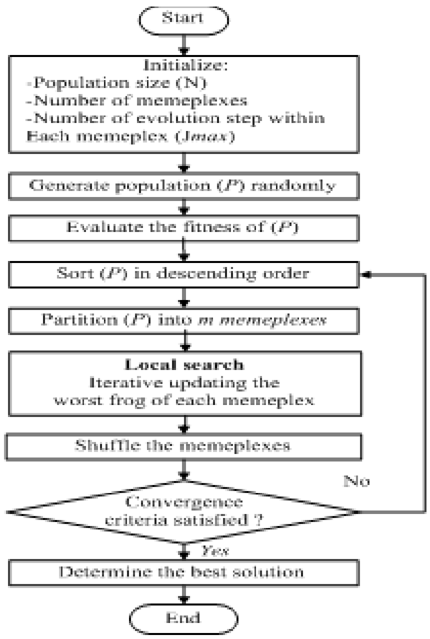


Figure 1. SFLA flow chart

The SFL algorithm is described in details as follows. First, an initial population of N frogs $P=\{X_1,X_2,...,X_N\}$ is created randomly. For S-dimensional problems (S variables), the position of a frog i^{th} in the search space is represented as $X_i=(x_1,x_2,\ldots,x_{is})^T$.

Afterwards, the frogs are sorted in a descending order according to their fitness. Then, the entire population is divided into m memeplexes, each containing n frogs (i.e. $N=m\times n$), in such a way that the first frog goes to the first memeplex, the second frog goes to the second memeplex, the m^{th} frog goes to the m^{th} memeplex, and the $(m+1)^{th}$ frog goes back to the first memeplex, etc. Let M_k is the set of frogs in the K^{th} memeplex, this dividing process can be described by the following expression:

$$M_k = \{X_{k+m(l-1)} \in P \mid 1 \le k \le n\}, (1 \le k \le m). \quad (10)$$

Within each memeplex, the frogs with the best and the worst fitness are identified as X_b and X_w, respectively. Also, the frog with the global best fitness is identified as X_g. During memeplex evolution, the worst frog X_w leaps toward the best frog X_b. According to the original frog leaping rule, the position of the worst frog is updated as follows:

$$D = r.(X_b - X_w) \quad (11)$$

$$X_w(new) = X_w + D, (\|D\| < D_{\max}), \quad (12)$$

Where r is a random number between 0 and 1; and D_{max} is the maximum allowed change of frog's position in one jump.

B. Particle swarm optimization algorithm

PSO is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995 [12-13]. The PSO algorithm is inspired by social behavior of bird flocking or fish schooling. The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, .the potential solutions, called particles, fly through the problem space by following the current optimum particles. Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas [14].

The standard PSO algorithm employs a population of particles. The particles fly through the n-dimensional domain space of the function to be optimized. The state of each particle is represented by its position $x_i = (x_{i1}, x_{i2}, ..., x_{in})$ and velocity $v_i = (v_{i1}, v_{i2}, ..., v_{in})$, the states of the particles are updated. The three key parameters to PSO are in the velocity update equation (13). First is the momentum component, where the inertial constant w, controls how much the particle remembers its previous velocity [14]. The second component is the cognitive component. Here the acceleration constant C_1, controls how much the particle heads toward its personal best position. The third component, referred to as the social component, draws the particle toward swarm's best ever position; the acceleration constant C_2 controls this tendency. The flow chart of the procedure is shown in Fig. 2.

During each iteration, each particle is updated by two "best" values. The first one is the position vector of the best solution (fitness) this particle has achieved so far. The fitness value $p_i = (p_{i1}, p_{i2}, ..., p_{in})$ is also stored. This position is called pbest. Another "best" position that is tracked by the particle swarm optimizer is the best position, obtained so far, by any particle in the population. This best position is the current global best $p_g = (p_{g1}, p_{g2}, ..., p_{gn})$ and is called gbest. At each time step, after finding the two best values, the particle updates its velocity and position according to (13) and (14).

$$v_i^{k+1} = wv_i^k + c_1 r_1 (pbset_i - x_i^k) + c_2 r_2 (gbset_k - x_i^k) \quad (13)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (14)$$

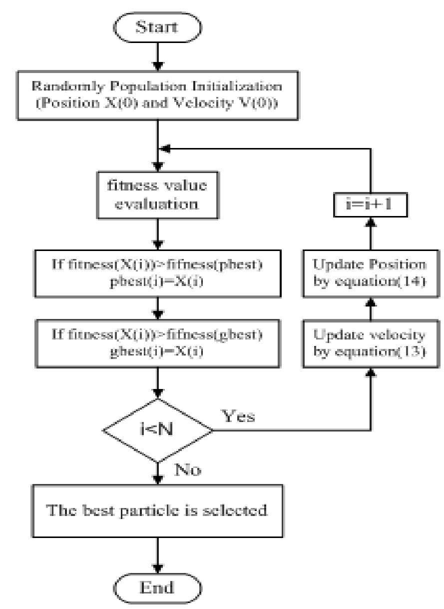


Figure 2. PSO flow chart

© 2012 ACEEE
DOI: 01.IJNS.03.02.72

V. SIMULATION RESULTS

The proposed algorithms applied to the IEEE 45 bus system. This system has 44 sections with 16.97562MW and 7.371194MVar total load as shown in Figure 2. The original total real power loss and reactive power loss in the system are 2.05809MW and 4.6219MVR, respectively.

Fig. 4 shows the convergence of proposed SFL and PSO algorithms for different number of capacitors. It is observed that the variation of the fitness during both algorithms run for the best case and shows the swarm of optimal variables. The improvement of voltage profile before and after the capacitors installation and they're optimal placement is shown in Figure 5. According to tables 1- 4 it is observed that the ratio of losses reduction percentage to the total capacity of capacitors which is one of the capacitors economical indicators. Also by comparing the voltage profile curves in the four cases with the curve before capacitors installation, it is observed that the voltage profile in the four cases is improved.

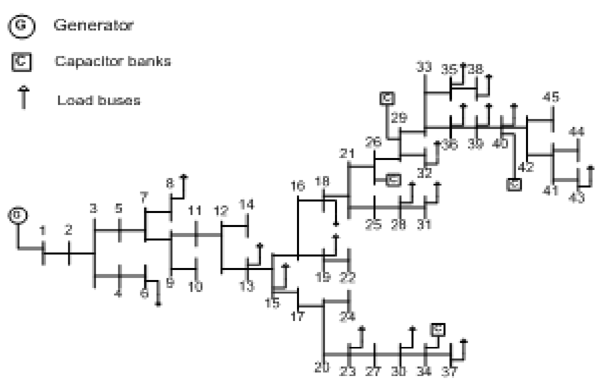
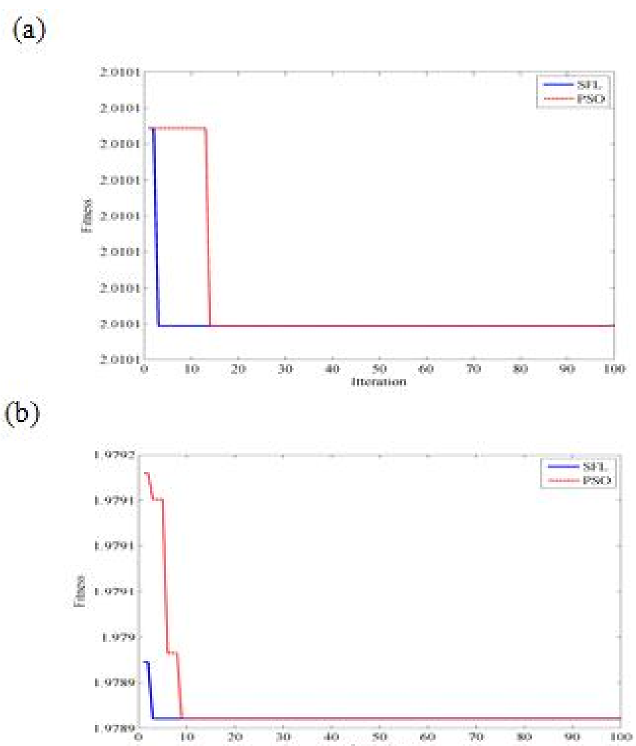


Figure 3. The IEEE 45 bus radial distribution system



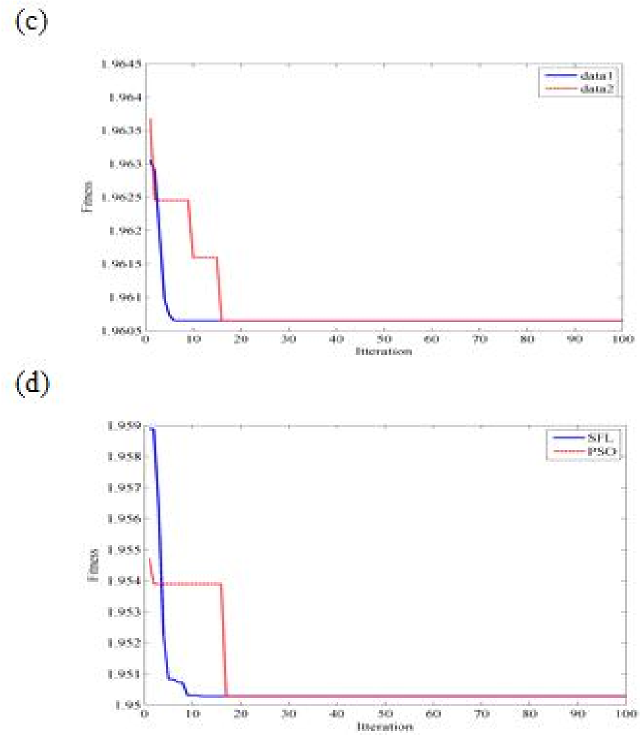


Figure 4. Convergence of the optimization of algorithms. (a). With 1 capacitor, (b). With 2 capacitors, (c). With 3 capacitors, (d). With 4 capacitors

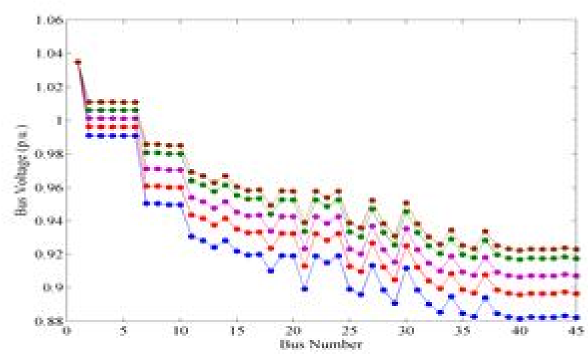


Figure 5. Bus voltage before and after capacitor Installation with SFL and PSO algorithms

VI. CONCLUSION

In this paper, the shuffled frog leaping (SFL) algorithm and particle swarm optimization (PSO) algorithm for optimal placement of multi-capacitors is efficiently minimizing the total real power loss satisfying transmission line limits and constraints. With comparing results and application of the two algorithms we should say that as it is observed the acquired voltage profile of the result of SFL algorithm is better than PSO algorithm. However the main superiority of this algorithm is in acquiring the best amount. Because SFL algorithm find the correct answer in the first repeating that are done to be sure of finding the best correct answer and the probability of capturing in the local incorrecting answers is very low. Also it is worthy or mentions that the time of performing this algorithm is faster. Finally we can say that SFL as compared with PSO is more efficient in this case.

© 2012 ACEEE
DOI: 01.IJNS.03.02.72

ACEEE

TABLE I.
OPTIMAL CAPACITOR PLACEMENT FOR 1 CAPACITOR WITH SFL AND PSO ALGORITHMS

Method	Capacitor Size (MVAR)	Bus No	Losses Without Capacitor		Losses With Capacitor	
			MW	MVAR	MW	MVAR
SFLA	1*1.2	12	2.058	4.6219	2.0101	4.5160
PSO						

TABLE II.
OPTIMAL CAPACITOR PLACEMENT FOR 2 CAPACITORS WITH SFL AND PSO ALGORITHMS

Method	Capacitor Size (MVAR)	Bus No	Losses Without Capacitor		Losses With Capacitor	
			MW	MVAR	MW	MVAR
SFLA	2*1.2	9	2.058	4.6219	1.9789	4.4476
PSO		12				

TABLE III.
OPTIMAL CAPACITOR PLACEMENT FOR 3 CAPACITORS WITH SFL AND PSO ALGORITHMS

Method	Capacitor Size (MVAR)	Bus No	Losses Without Capacitor		Losses With Capacitor	
			MW	MVAR	MW	MVAR
SFLA	3*1.2	7	2.058	4.6219	1.9606	4.4092
		9				
PSO		11				

TABLE IV.
OPTIMAL CAPACITOR PLACEMENT FOR 4CAPACITORS WITH SFL AND PSO ALGORITHMS

Method	Capacitor Size (MVAR)	Bus No	Losses Without Capacitor		Losses With Capacitor	
			MW	MVAR	MW	MVAR
SFLA	4*1.2	5	2.058	4.6219	1.9503	4.3870
		7				
		9				
PSO		11				

REFERENCES

[1] Y. H. Song, G. S. Wang, A. T. Johns and P.Y. Wang, "Distribution network reconfiguration for loss reduction using Fuzzy controlled evolutionary programming," *IEEE Trans. Gener., trans., Distri.*, Vol. 144, No. 4, July 1997

[2] J. V. Schmill, "Optimum Size and Location of Shunt Capacitors on Distribution Feeders," *IEEE Trans.on PAS*,vol.84,pp.825-832,Sept.1965.

[3] H. Dura "Optimum Number Size of Shunt Capacitors in Radial Distribution Feeders: A Dynamic Programming Approach", *IEEE Trans. Power Apparatus and Systems*, Vol. 87, pp. 1769-1774, Sep 1968.

[4] J. J. Grainger and S. H. Lee, "Optimum Size and Location of Shunt Capacitors for Reduction of Losses on Distribution Feeders," *IEEE Trans. on PAS*, Vol. 100, No. 3, pp. 1105- 1118, March 1981.

[5] J.J.Grainger and S.Civanlar, "Volt/var control on Distribution systems with lateral branches using shunt capacitors as Voltage regulators-part I, II and III," *IEEE Trans. PAS*, vol. 104, No.11, pp.3278-3297, Nov.1985.

[6] M. E Baran and F. F. Wu, "Optimal Sizing of Capacitors Placed on a Radial Distribution System", *IEEE Trans. Power Delivery*, vol. No.1, pp. 1105-1117, Jan. 1989.

[7] M. E. Baran and F. F. Wu, "Optimal Capacitor Placement on radial distribution system," *IEEE Trans. Power Delivery*, vol. 4, No.1, pp. 725- 734, Jan. 1989.

[8] Sundharajan and A. Pahwa, "Optimal selection of capacitors for radial distribution systems using genetic algorithm," *IEEE Trans. Power Systems*, vol. 9, No.3, pp.1499-1507, Aug. 1994.

[9] Baghzouz. Y and Ertem S, "Shunt capacitor sizing for radial distribution feeders with distorted substation voltages," *IEEE Trans Power Delivery*,

[10] I.O. Elgerd, Electric Energy System Theory: McGraw Hill., 1971.

[11] M.M.Eusuff and K.E.Lansey,"Optimization of water distribution network desighn using the shuffled frog leaping algorithm,"*J.water Resources Planning & Management*,vol.129(3),pp.210-225,2003.

[12] J.Kennedy, and R.C. Eberhart, "Particle swarm optimization.", proc. Int. conf. on neutral networks, Perth, Australia, pp.1942-1948, 1995.

[13] R. C. Eberhart, and Y. Shi, "Particle swarm optimization: developments, applications and resources.", *Proc. Int. Conf. on evolutionary computation, Seoul, Korea*, pp. 81-86, 2001.

[14] Y. Shi and R. Eberhart, "A modified particle swarm optimizer, " *Proc. Int. Conf. on Evolutionary Computation, Anchorage, AK, USA*, pp. 69-73, 1998.

© 2012 ACEEE
DOI: 01.IJNS.03.02.72